"Wall Hack AR": AR See-through system with LiDAR and VPS

Ryudai Kimura* and Michael *Cohen***

Spatial Media Group, Computer Arts Lab; University of Aizu; Aizu-Wakamatsu, Fukushima; Japan

Abstract. We propose a method to realize the experience of seeing through an occluding wall by using a 3D scanner and an AR (Augmented Reality) device. This system uses the LiDAR (Light Detection And Ranging) scanner of the Apple iPad Pro as a 3D scanner, sending scanned spatial mesh data to the AR device of the experiencer in real-time. By using a VPS (Visual Positioning System) to also share the actual spatial position of the mesh, a receiving AR device can also draw the scan data registered at the correct position in the actual space. Placing the LiDAR scanner behind an occlusion allows an experience as if the user were seeing through a wall.

1 Introduction

1.1 Background

The experience of seeing through walls and other occluded objects is often used in science fiction movies, games, and animation. For example, in the futuristic first-person shooter (FPS) game "Apex Legends" [1], a drone and some kinds of sensors can be used to scan the other side of an obstacle, and scanned terrain and characters can be highlighted on a HUD (Head-Up Display). The experience of seeing through a wall can leave a deep impression on the experiencer as entertainment content, and to increase the convenience of working in places with a lot of occlusions.

1.2 Related work

A familiar example of working in a place with many obstructions is driving a vehicle. The driver's view is obstructed by the vehicle's walls and seats. Tachi et al. [2] made the interior walls appear transparent by projecting images taken by an exterior camera onto the interior walls covered with retroreflective material. "F-35 Lightning II Cockpit Vision" [3] projects a panoramic image inside a pilot's helmet by stitching together images from infrared cameras outside the fighter, allowing the pilot to see through the walls of the aircraft from the cockpit. In 2004, a see-through system in a building using mobile AR devices was proposed [4], and the need for a mechanism to track the location of AR devices was noted. In addition, olde Scholtenhuis et al. [5] and Ortega et al. [6] worked on the visualization of underground infrastructure using AR see-through systems. Zhang et al. [7] improved performance of manual assembly in Vision Blind Areas (VBAs) in an industrial setting with their see-through system. This system uses a data glove and HoloLens to visualize the position of human hands and machine parts in VBAs. As observed by Bane and Hollerer [4], to realize an AR see-through system, it is necessary to have a mechanism to track the location of the device. This is because scan data cannot be displayed on an AR device unless the spatial location of the scan data and the AR device can be determined and aligned. The studies of olde Scholtenhuis et al. [5] and Ortega et al. [6] used GPS, while Zhang et al. [7] used AR markers. However, GPS used in smartphones is not accurate enough for positioning, especially indoors. A more accurate localization method than GPS is Visual Positioning System (VPS) [8]. A VPS system estimates position and orientation based on camera imagery. Also, there is no need to install AR markers.

* bigdra50@gmail.com

** mcohen@u-aizu.ac.jp

In recent years, VPS that can be easily used with smartphones have been released, making it easier to develop AR applications that connect to real space.

1.3 Goal of this research

In this research, we propose an AR See-Through System using a LiDAR scanner and VPS. Using LiDAR, we can perform real-time 3D scanning and transfer the scanned data to the AR devices in an adjoining room. By positioning the scan data in VPS, it can be displayed in the correct position in real space. This allows the experience as if seeing through a wall.

2 Resources

2.1 Hardware

2.1.1 Apple iPad Pro

The iPad Pro [9], as seen in the top left at Fig. 1, is a tablet computer by Apple Inc. It is equipped with a Time of Flight (ToF) LiDAR scanner. The LiDAR can scan objects in 3D by emitting pulsed or modulated light signals and measuring the time difference between the returning wavefronts. This process is performed in full frame resolution and in real-time, independent of the texture and detail of the scene [10].

2.1.2 NReal Light

NReal Light [11], as seen towards the bottom right of Fig1, is an optical see-through MR eyewear device by NReal Ltd. It supports SLAM, 6-DoF, and hand tracking. The glasses are not equipped with a processing unit, but are connected to an Android device via USB tether and use the power and processing power of the Android device.

2.1.3 Google Android Smartphone

A Google Android smartphone is used for processing NReal Light as shown on the top right of Fig. 1. NReal Light support is limited in each country, and for this project we used Samsung Galaxy Note20 Ultra 5G [12].

2.2 Software

2.2.1 Unity

Unity [13] is a development platform for creating 2D and 3D multiplatform games and interactive experiences by Unity Technologies as shown on the bottom at Fig. 1. It has a full range of frameworks for AR application development.

2.2.2 AR Foundation

AR Foundation [14] is an AR development framework for Unity. It abstracts the core functions of different AR frameworks, such as Apple ARKit [15] and Google ARCore [16], into a unified interface. By using AR Foundation's `ARMeshManager` class, LiDAR scanning and mesh generation according to the actual terrain can be performed.

2.2.3 Immersal

Immersal [17] is a visual positioning system by Immersal Inc seen towards the lower center of Fig. 1. A Unity plugin for iOS and Android is provided and enables fast visual positioning with low system overhead. Positioning works both offline on device and online in the cloud. In order to perform positioning, mapping of the space is necessary in advance. Map construction works by finding corresponding feature points in multiple images from different viewpoints and triangulating those feature points to estimate the 3D structure of the scene [18]. Since this process is done in the Immersal cloud, all the user has to do is to capture pictures of the target environment from different perspectives and upload them to the cloud. The

Mapper application provided by Immersal can be used for mapping. When mapping is successful, mapped space point clouds and textured meshes become available for download. These point clouds are used for localization.

2.2.4 **Mirror**

Mirror [19] is a high level networking library for Unity, compatible with different low level transports seen the upper center of Fig. 1. It has been adopted by many online games. In this project, we used it to transfer mesh data between iPad Pro and NReal Light (Android device).

3 Implementation

3.1 Overview

The system architecture is shown in Fig. 1. This system operates as follows:

1. Preparation
   a. Create with Immersal a feature point map of the site to be used.
   b. Load the map into the “Wall Hack AR” app of iPad Pro and NReal Light.
   c. NReal Light and iPad Pro start localizing with VPS to identify the position of each device.
   d. Start the UDP connection with NReal Light as the host and iPad Pro as the client.
2. Generating a spatial mesh
   a. The iPad Pro scans the 3D space with LiDAR, generates a mesh of the space, and transfers the mesh shape (vertex coordinates and triangles) to NReal Light.
   b. NReal Light recreates a mesh based on the transferred shape.
   c. The mesh generated in 2b is overlaid on NReal Light’s view.
3. Spatial localization of mesh
   a. The iPad Pro deploys the mesh data generated in 2a into the volumetric map of Immersal. This is necessary for the next step.
   b. The iPad Pro obtains the spatial position and orientation of the mesh data generated in 2a from Immersal’s VPS.
   c. The iPad Pro transfers the spatial position and orientation obtained in 3b to NReal Light.
   d. NReal Light applies the spatial position and orientation transferred in 3c to the generated mesh.
4. Repeat steps 3–4 for duration of session.

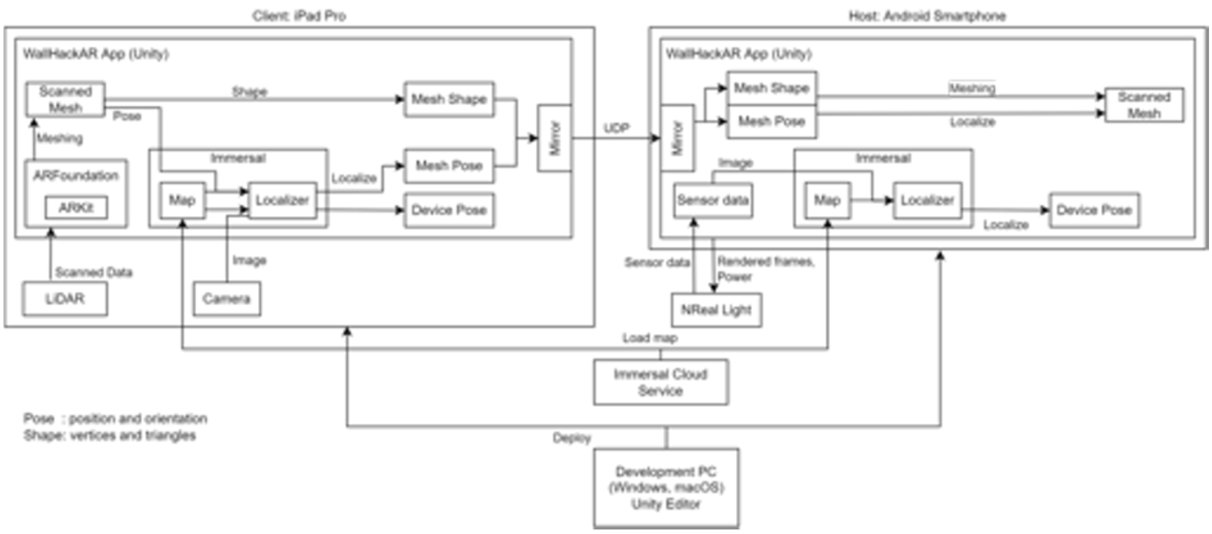

Figure 1: System architecture

3.2 Mapping

For localization in Immersal's VPS, it is necessary to create a volumetric map of the experience area in advance. For mapping, we used Immersal Mapper published by Immersal. In this case, we created a map including the inside and outside of the laboratory, as shown in Fig. 2. Immersal's VPS can estimate a device's spatial location from the feature points of this map and images captured by the camera at runtime. Once mapped, the point cloud data is uploaded to the Immersal cloud service. The created point cloud map is downloaded via Immersal Cloud Service at runtime.

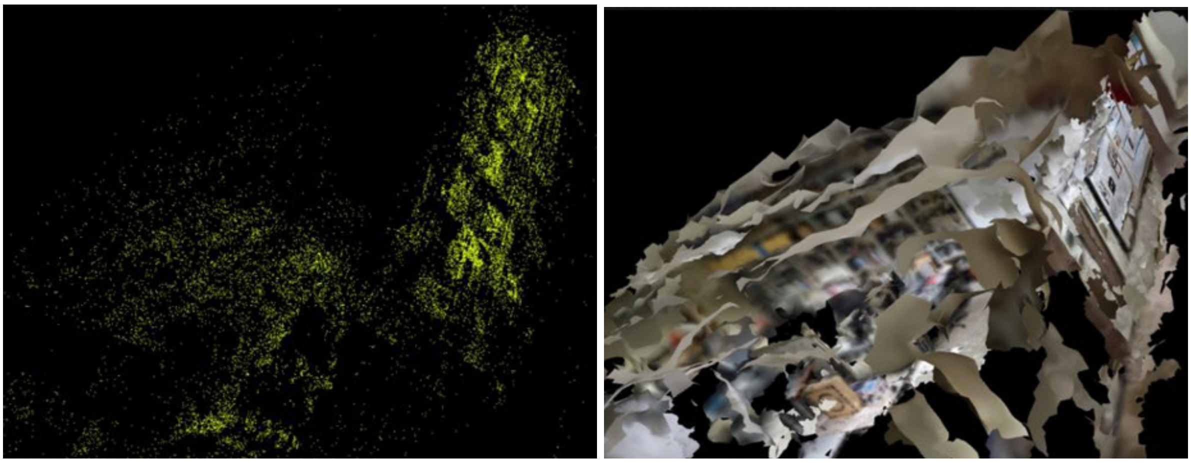

Figure 2: Point cloud map and textured map

3.3 Generating a spatial mesh

The iPad Pro's LiDAR 3D scanning and meshing features are exposed through ARFoundation's ARMeshManager class. From the mesh data generated by the iPad Pro, only the minimum information necessary to construct the mesh is extracted and transferred to NReal Light via UDP (Table 1). NReal Light generates a mesh object each time it receives mesh data and displays it as an overlay in the view, as seen in Fig. 3. This scanning, data transfer, and meshing are performed in real-time, but the size of the mesh data is so large that transferring the entire dataset every frame would cause a processing failure. In this implementation, data transfer is performed at an interval of once every 60 frames. In addition, data transfer is not performed for frames in which there is no change in the shape of the scanned mesh. Change in mesh shape is determined by the change in the number of mesh vertices.

Table 1: Mesh Shape Message

Instance Id	int
Vertices	Vector3[]
Triangles	int[]

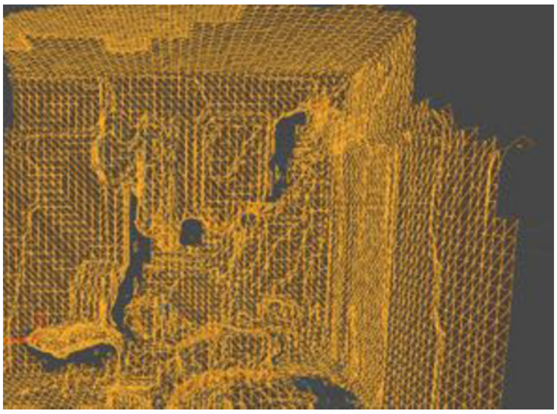

Figure 3: Generated spatial mesh

3.4 Localization of mesh

LiDAR-generated meshes on the iPad Pro are positioned in real space on the iPad Pro by functions of ARKit. However, this position is in the app's world coordinate system based on the device's position when the app is launched. Similarly, the app on the NReal Light side uses world coordinates based on the position of the device at the time of launching the app on the NReal Light side. Since each device has a different world coordinate system as a reference, the mesh location and orientation cannot be used directly. Therefore, we are using localized positions and orientations from Immersal's VPS. These positions are based on the point cloud map of the experience area that we created in advance, and we can use the same positions and orientations on each device. Therefore, on the iPad Pro, localized location and orientation obtained from Immersal's VPS are transferred to NReal Light (Table 2). By applying these positions to generated mesh objects, NReal Light can align the meshes to their actual spatial positions.

Table 2: Mesh Pose Message

Instance Id	int
Position	Vector3
Orientation	Quaternion

3.5 Hand tracking

In order to make the experience more immersive, NReal Light uses hand tracking for manipulation (Fig. 4). Hand tracking is used to control the UI for loading the map and hosting the server, and to switch between rendering the meshes by gestures. Meshes are displayed when hands are close to each other, and hidden when hands are separated.

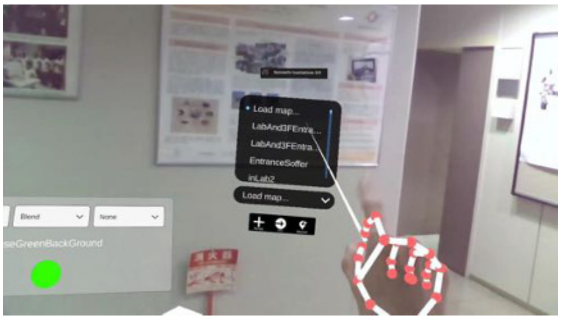

Figure 4: Map-loading panel

4 Deployment

4.1 Demonstration

Validation was conducted inside and outside rooms of the University of Aizu's Computer Arts Lab. Participants wore the NReal Light to experience seeing through the wall. The iPad Pro for 3D scanning can be set up beforehand in an adjacent room, or another participant in that room can walk around while holding it. First, the map list is opened in the Mapper tab on the NReal Light and iPad Pro app screen, selecting a map that was created in advance. Then, the corresponding map is loaded and localization by Immersal's VPS begins (Fig. 5). The localization status is always displayed in the center of the view in the form of "success count/localization count." Next, the NReal Light user can start the host mode from the Server tab and wait for the iPad Pro user to join the session.

iPad Pro users can connect to a server from the Join tab. From the join tab, a user can either join by specifying the IP address directly, or by searching for servers running in the same network and selecting the found server to join. After each device is connected, the iPad Pro starts 3D scanning and sends scan data to NReal Light. When the NReal Light user puts his or her hands close together, the mesh data is displayed.

Figure 5: Demonstration participants

4.2 Results

The iPad Pro transmitted the scanned data in real-time, and NReal Light could confirm that the scanned mesh data was displayed in real-time at the correct location in the actual space. Also, as shown in Fig. 6, by running each device in a different room, we realized an experience as if one could look through walls.

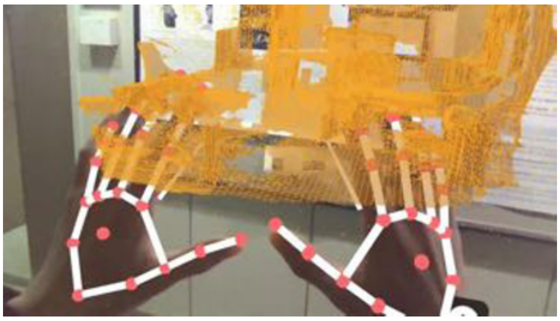

Figure 6: Perspective of seeing through the wall

4.3 Discussion

The data scanned in real-time by LiDAR was transferred to NReal Light, and accurate positioning by VPS was also successfully achieved. But there is some room for improvement. First, the amount of data to be transferred becomes too large when the scanning area is enlarged or when scanning for a long time, and the communication is interrupted. Furthermore, it was difficult to get a sense of distance from the mesh when looking through half of the view from the wall.

4.4 Future work

To solve the problem of data transfer, we will add functions such as splitting the data when it is large, and consider other communication methods. If the data transfer problem is improved, it is assumed that 3D scanning using multiple LiDARs simultaneously will be possible. Moreover, If the scanning device can record spatial sound, it will be possible to stream the spatial sound and hear through walls.

Acknowledgement

We express appreciation to Genji Kaiyu for his help with the experiments.

References

1. Respawn Entertainment. “Apex Legends.” PC [Game] (Update 1.1. 3) (2019)
2. S. Tachi, M. Inami, and Y. Uema, “The transparent cockpit,” IEEE Spectrum, vol. 51, no. 11, pp. 52–56, (2014)
3. M. Skaff, “F-35 Lightning II Cockpit Vision 2010-01-2330,”
4. R. Bane and T. Hollerer, “Interactive tools for virtual x-ray vision in mobile augmented reality,” Third IEEE and ACM Int. Symp. on Mixed and Augmented Reality, pp. 231–239, (2004)
5. L. olde Scholtenhuis, S. Zlatanova, X. den Duijn, A.M. Ntarladima, and E. Theocharous, “Spying the underground: visualizing subsurface utilities’ location uncertainties with fuzzy 3D,” SPOOL, vol.4, no.2, pp.61–64, (2017)
6. S. Ortega, J. Wendel, J.M. Santana, S.M. Murshed, I. Boates, A. Trujillo, A. Nichersu, and J.P. Suárez, “Making the invisible visible–strategies for visualizing underground infrastructures in immersive environments,” ISPRS Int. J. of Geo-Information, vol. 8, no. 3, p. 152, (2019)
7. S. Zhang, W. He, S. Wang, S. Feng, Z. Hou, and Y. Hu, “An AR-Enabled See-Through System for Vision Blind Areas,” Int. Conf. on Human Computer Interaction, pp. 206–213, Springer, (2021)

8. S. Treuillet and E. Royer, "Outdoor/indoor vision based localization for blind pedestrian navigation assistance," Int. J. Image Graphics, vol. 10, pp. 481–496, 10 (2010)
9. "Apple Inc. iPad Pro (2nd generation)." https://support.apple.com/kb/SP814/ (Accessed on 01/21/2022).
10. S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–7, (2008)
11. "NReal Light." https://www.nreal.ai/light/ (Accessed on 01/21/2022).
12. "Samsung Galaxy Note20 Ultra 5G." https://www.samsung.com/us/smartphones/galaxy-note20-5g/specs/ (Accessed on 01/21/2022).
13. "Unity." https://unity.com/ (Accessed on 01/21/2022).
14. "ARFoundation." https://unity.com/unity/features/arfoundation/ (Accessed on 01/21/2022).
15. "ARKit." https://developer.apple.com/augmented-reality/arkit/ (Accessed on 01/21/2022).
16. "ARCore." https://developers.google.com/ar/ (Accessed on 01/21/2022).
17. "Immersal." https://immersal.com/ (Accessed on 01/21/2022).
18. "Immersal's tutorials how-to-map." https://immersal.gitbook.io/sdk/tutorials/how-to-map/ (Accessed on 01/21/2022).
19. "Mirror." https://github.com/vis2k/Mirror/ (Accessed on 01/21/2022)